PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : H04L 29/12, G05B 19/418 — A1

(11) International Publication Number: WO 00/41377

(43) International Publication Date: 13 July 2000 (13.07.00)

(21) International Application Number: PCT/US99/23658

(22) International Filing Date: 13 October 1999 (13.10.99)

(30) Priority Data:
09/224,196 30 December 1998 (30.12.98) US

(71) Applicant: SCHNEIDER AUTOMATION INC. [US/US]; One High Street, North Andover, MA 01845 (US).

(72) Inventors: NAISMITH, Ronald; 42 Huckleberry Lane, North Andover, MA 01845 (US). TANZMAN, Allan; 37 Wessex Road, Newton Centre, MA 02459 (US). PAPDOPOULOS, A., Dean; 11 Rooks Run, Groton, MA 01450 (US). SWALES, Andrew, G.; 10 Woodvue Road, Windham, NH 03087 (US). METCALF, Orlando, P.; 475 Wood Lane, North Andover, MA 01845 (US).

(74) Agent: GOLDEN, Larry, I.; General Patent Counsel, Square D Company, 1415 South Roselle Road, Palatine, IL 60067 (US).

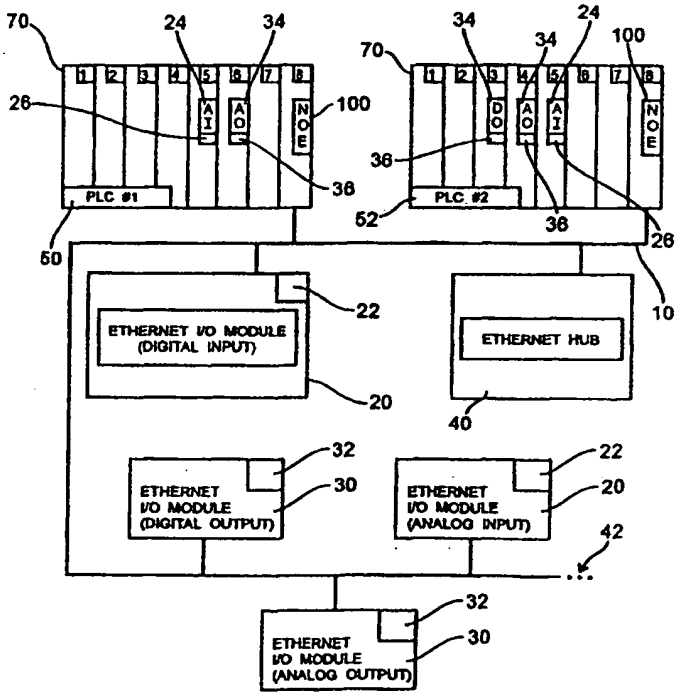
(81) Designated States: CA, MX, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.

(54) Title: INPUT/OUTPUT (I/O) SCANNER FOR A CONTROL SYSTEM WITH PEER DETERMINATION

(57) Abstract

The present invention is directed to an apparatus for communication with at least one device which resides on a standard communications network using a standard communications protocol. The apparatus has a scanner for scanning the device, a device scan table for storing data relating to the device, and a standard communications interfaces for interfacing between to the device scanner and the standard communications network using the standard communication protocol. The present invention is also directed to a device scanner for a first device located on a first node of a standard communications network. The device scanner is provided for scanning devices on the standard communications network and for identifying a second device on a second node of the standard communications network. The device scanner has an initiator for initiating a first communications command in a peer protocol format to the second node, a receptor for receiving from the second node a second communications command in the peer protocol format, in response to the first communications command, and an identifier for identifying the second device on the second node as a peer device. This apparatus and device can be used within a control system for monitoring input devices and for controlling output devices which reside on the standard communications network. The standard communications network can be an Ethernet network, and the standard communications protocol used therein can TCP using Modbus.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

Input/Output (I/O) Scanner for a Control System with Peer Determination

DESCRIPTION

Technical Field

The present invention generally relates to control systems. More specifically, the present invention relates to an apparatus and method for determining the type of a communication device using known communication protocols over standard networks, and for moving data, relating to input and output (I/O) devices within the control system, to and from a programable logic controller.

Background of the Invention

Within control systems, there has a need to make I/O devices and modules, programmable logic controllers (PLCs), and other devices capable of being used on standard communications protocols, such as Ethernet, TCP/IP, and others. This includes the ability to interface a proprietary communications protocol with standard protocols. Previous I/O scanners within such devices, typically used proprietary control networking protocols. Using proprietary control networking protocols, resulted in high installation costs, low ease-of-use, and compatibility problems with other devices/systems used in control systems, such as in factory automation applications. U.S. Patent Nos. 5,159,673 (Sackman et al.), 4,992,926 (Janke et al.), 4,897,777 (Jamke et al.), 5,245,704 (Weber et al.), 4,937,777 (Flood et al.), 5,307,463 (Hyatt et al.), and /or 5,805,442 (Crater et al.) provide some background and context for the present invention.

The present invention is directed to solving the above mentioned and other problems.

Summary of the Invention

The present invention is an apparatus for communication with at least one device which resides on a standard communications network, such as an Ethernet network, using a standard communications protocol, such as TCP using Modbus. The apparatus has a scanner for scanning the device, a device scan table for storing data relating to the device, and a standard communications interface for interfacing between the device scanner and the standard communications network using the standard communication protocol.

The present invention is also a device scanner for a first device located on a first node of a standard communications network. The device scanner is provided for scanning devices on the standard communications network, and for identifying a second device on a second node of the standard communications network. The device scanner has an initiator for initiating a first communications command in a peer protocol format to the second node, a receptor for receiving from the second node a second communications command in the peer protocol format, in response to the first communications command, and an identifier for identifying the second device on the second node as a peer device. This apparatus and device can be used within a control system for monitoring input devices and for controlling output devices which reside on the standard communications network, as will be described in detail below.

Brief Description of the Drawings

FIG. 1 is a functional block network diagram of an example of a control system network over Ethernet of the present invention.

FIG. 2 is a functional block diagram of one embodiment of the present invention and the connections to other portions of the control system of the present

invention.

FIG. 3 is a state diagram of the control states of a scanner of present invention.

FIG. 4 is functional block and timing diagram of one embodiment of peer device determination of the present invention.

FIGs. 5A-5G are a detailed flowchart of the one embodiment of software code running on one embodiment of the scanner of the present invention.

Detailed Description

While this invention is susceptible of embodiment in many different forms, there is shown in the drawings and will herein be described in detail preferred embodiments of the invention with the understanding that the present disclosure is to be considered as an exemplification of the principles of the invention and is not intended to limit the broad aspect of the invention to the embodiments illustrated.

Referring to Figures 1 and 2, one embodiment of the present invention is an apparatus for communication with at least one device 22, 26, 32, and/or 36, which resides on a standard communications network 10, such as an Ethernet network, using a standard communications protocol, such as TCP. The apparatus has a scanner 110, which resides within the element labeled as the "NOE" 100 in Figure 1. "NOE" stands for Network Options Ethernet module or card. In the embodiment/example shown in Figure 1, the NOE is a communications card which fits into a backplane 70 having several slots (eight in Figure 1) for various cards/modules, such as local input modules 24 and local output modules 34, within the control system. One embodiment of the present invention is implemented with software (also referred to as firmware, or Exec), which runs on the NOE module having a microprocessor and memory. However, the present invention can be implemented in various different ways, such as having the scanner 110 residing on the PLC 50, 52 itself, depending on the implementation of the invention. The

device scanner is provided for scanning the devices 22, 26, 32, 36, both locally (located on its own backplane) or remotely (over the standard communications network, shown as an Ethernet network in the Figures). A device scan table 120 is provided for storing data relating to the devices 22, 26, 32, 36. A standard communications interface 140, such as a TCP/IP stack with an Ethernet driver, is provided for interfacing between the device scanner 110 and the standard communications network 10 using the standard communication protocol, such as TCP. The Ethernet network embodiment of the present invention uses Ethernet as the device level 10 network. This provides a fast, flexible, and convenient way of interconnecting the I/O devices of different PLCs or I/O modules to the PLCs.

The device scan table 120 includes several parameters which can be used by the scanner 110 to communicate with the devices 22, 26, 32, 36. A listing of the parameters in one specific embodiment of the present invention are provided in a chart further below. Some of these parameters within the scan table 120 are as follows: a first scan parameter is provided for indicating the number of devices 22, 26, 32, 36 to be scanned by the device scanner 110. A second scan parameter is provided for indicating a device type. A third scan parameter is provided for indicating where to retrieve and store data for the devices 22, 26, 32, 36. A fifth scan parameter is provided for indicating the length of the stored data and the retrieved data.

Within the control system, the programmable logic controllers (PLC) 50, 52 communicate between themselves and with other devices using a PLC communication protocol. One such protocol is Modbus, a known protocol. In the present embodiment, the PLC communications protocol is communicated over the standard communications protocol, such as TCP. Detailed information on the Modbus protocol and TCP and TCP/IP can be found on the Internet at www.modicon.com, and other locations, including documentation listed as "Open

Modbus/TCP Specification," which is hereby incorporated by reference into the present specification. The standard communications network 10 provides communication between the device scanner 110 and the remote devices 22, 32 and modules 20, 30. Within Figure 1, element 42 depicts that many other devices can be connected to the standard communications network 10, as is well known. As indicated above, in one specific embodiment of the present invention, the I/O scanner 110 scans the I/O devices by using the Modbus™ protocol (from Schneider Automation, Inc.) over TCP/IP. In this embodiment, the I/O scanner 110 uses the read registers, write registers, and the read/write registers' Modbus commands to move data to and from the PLC memory. This will allow specific types of PLCs within the control system to efficiently transfer repetitive data to Ethernet modules, other types of PLCs, and any other Ethernet TCP/IP device that supports the MB (Modbus) protocol.

The control system shown in Figure 1 can have numerous PLCs 50, 52. Each PLC 50, 52 typically has a microprocessor and memory (PLC memory such as Random Access Memory - RAM), with software or firmware running therein. Within the embodiment shown in Figures 1 and 2, the PLCs 50, 52 each have PLC memory that includes a configuration table. The PLC memory configuration table can have the same parameters listed within the scan table 120 listed above and/or listed within the table of parameters further below. In the embodiment using a configuration table within the PLC memory, the parameters within the configuration table of the PLC memory are read into the scan table 120 upon start-up of the NOE 100 and/or device scanner 110. However, other embodiments of the present invention can have the parameters read into the scan table 120 by other means, such as through a web page (accessible through the world wide web (www)) located on the NOE itself. This type of NOE could generally be called a web-embedded server module. Alternatively, the parameters could be placed into

the scan table through a user creating/editing a file on the user's personal computer, and the user could send the file to the NOE using a File Transfer Protocol (FTP) or some other transfer means from a remote location.

In the embodiment shown in Figures 1 and 2, the PLCs 50, 52 are adapted to communicate with the local input and output (I/O) devices 26, 36 through the back plane 70, and with the remote input and output (I/O) devices 22, 32 through the back plane 70, the NOE 100, the network 10, and the I/O modules 20, 30 on the network 10. The NOE 100 in the embodiment in Figures 1 and 2 is adapted for communication with the PLC 50, 52, the local input and output devices 26, 36, the Ethernet network, 10 and the remote input and output devices 22, 32. As stated above, and as will described in detail further below, the NOE server module 100 has a scanner 110 for scanning the input and output devices, an I/O scan table 120 for storing real time and other information for the input and output devices, a standard communication protocol interface 142 and a standard communication network driver 144 for interfacing between the I/O scanner 110 and the standard communication network 10 using the standard communication protocol. The standard communications network driver 144 can be a commercially available AM79C961 Driver.

In the embodiment shown in Figures 1 and 2, the NOE module 100 also has a real time operating system for running the various tasks on the NOE, including the "IO scan task" or scanner 110. Commercially available operating systems can be used, such as the PSOS real time operating system manufactured by Integrated Systems, Inc. of Sunnyvale, California. Information on the PSOS real time operating system is available from this company and/or on the Internet at www.isi.com. One preferred real time operating system that can be used is VXWORKS provided by a company named Wind River Systems, Inc. of Alameda, California. VXWORKS has been used within the QUANTUM product

line of Schneider Automation, Inc., the Assignee of the present invention. Some of the embodiments of the present invention do not need a real time operating system, such as the "M1" product line of Schneider Automation, Inc., in which case, the firmware runs on a processor without the assistance of the real time operating system.

In one embodiment, the operation of the scanner 110 is configured using a commercially available panel software applications, such as Modsoft™ or Concept™. The panel software is used to input the information about I/O devices, which are to be scanned by the NOE, that is, to be written to and read from.

In a particular embodiment of the present invention, the scan table 120 includes: the number of 16-bit words that the device accepts as input or produces as output; the source or destination address in the controller memory space (referred to as OX, IX, 3X, or 4X Registers); the timeout value for a device, which is the amount of time which is allowed to elapse before a device is considered to be unhealthy; a flag to indicate what to do with input data when a device has stopped responding, the two choices are HOLD or ZERO. This implementation produces the IP address of the device from its Modbus address as follows:

Device ' s IP Address: AA.BB.CC.MB

AA.BB.CC is the 1st 3 octets of this NOE' s IP address, and MB is the Modbus Address which has been entered with the panel software. A further embodiment includes the ability to directly enter the remote I/O devices, IP addresses, as well as the device types.

A TCP connection shall be reserved for each I/O device in the scan list, until the maximum of 128 devices are reached. As used within this specification, FC 3 means function code 3, which is a MB (Modbus) read register message. Likewise,

FC 16 means function code 16, is also an MB write register message, and FC 23 means function code 23, which is an MB read/write message. The preferred embodiment of the present invention is capable of supporting Peer Cop, the name of a particular control system arrangement of Schneider Automation, Inc., which is described in U.S. patent application serial number 60/078,223, entitled Communications System For A Control System Over Ethernet And IP Networks and Communications Interfaces for Such Systems," which is hereby incorporated by reference into the present specification. Additional information on "peer cop" is also available on the Internet at www.modicon.com, which is hereby incorporated by reference herein. In this embodiment, the user can use the peer cop input screens in the panel software to configure the NOE, including the I/O scan table, although there are other ways to configure the NOE, as are described within the present specification.

The general setup and flow of the I/O scanner software/firmware is shown in Figure 3. The following next state transition table indicates the next states of the next state diagram of Figure 3.

Current State	Next State	Trigger Number	Trigger
IOScanEmpty	IOScanStarting	1	ExitDim
IOScanStarting	IOScanNewCfg	2	New Cfg. From controller and Controller is running
IOScanRunning	IOScanStopped	3	Controller Not Running
IOScanNewCfg	IOScanStopped	6	Controller Not Running
IOScanStarting	IOScanStopped	4	Controller Not Running
IOScanStopped	IOScanStarting	5	ExitDim
IOScanNewCfg	IOScanReadCfg	7	IOScanState still IOScannewCfg

IOScanReadCfg	IOScanRunning	8	IOScanState still IOScanReadCfg and Open IOScan connections

In one embodiment, the I/O scanner 110 utilizes the event flag capability of the PSOS real-time operating system (RTOS). This allows up to 16 user defined events to be posted to a task. The I/O scanner interface to the backplane (BP) driver is via a call-back function. The BP driver calls the call back function to interface to the I/O scanner 110. The call back function posts the appropriate event to the I/O scanner 110. The I/O scanner 110 runs in a forever loop, checking for events. If an event is posted, the I/O scanner 110 carries out the appropriate function. The event flag communication within the PSOS RTOS does not provide the full functionality of the message queue communication method, but requires much less system resources. Event flags do not need to be queued in this embodiment.

The I/O scanner 110 utilizes the services of a client task 160 to implement I/O scanning. The client portion or task 160 is the portion of the NOE 100 which handles the "client" tasks. For example, the user can write a control program for the PLC 50 which, as a part of the operation of the program on the PLC, the PLC will send/receive a Modbus message to/from the backplane 70, and these types of messages will be handled by the client task 160 to/from TCP/IP stack 142 and Ethernet driver 144. Parameters are used to pass pointers for connection list and connection arrays.

With reference to the above next state transition table and Figure 3, there is no processing required during the IOScanEmpty state. The backplane driver initializes the I/O scan table 120 through existing or created configuration data located elsewhere, as has been described herein. During the IOScanNewCfg State the I/O scanner disables interrupts, and tests to see if the state is still IOScanNewCfg. If the state is IOScanNewCfg, the IO scanner 110 task changes the IO Scan state to

IOScanReadCfg, and re-enables interrupts. The I/O scanner 110 also performs the necessary housekeeping, to remove any previously used open connections. If the test for IOScanNewCfg fails, then the IO scanner 110 re-enables interrupts and exits. The IO scanner 110 copies the IO scan data structure to its own local variable to ensure that here will not be a contention issue with the BackPlane Driver trying to access the data at the same time.

Another aspect of present invention includes a "peer" determination portion that , among other things, was implemented to allow for versatility of the control system. As will be described in greater detail below, one particular embodiment of the present invention includes using two determinations to determine whether a device (in the scan list) is a peer: (1) Does the device understand the Modbus read/write register command (Function Code 23) and (2) does the information in the scan table 120 match the communication that the remote device is directing at this node? If these two conditions are met then the device is determined to be a "peer" device.

Referring to Figure 4, in view of Figures 1 through 3, in general, the present invention is a method for identifying a second device 210 on a second node of a standard communications network 10 from a first device 200 located on a first node of the standard communications network 10. The method first initiates from the first node a first communications command in a peer protocol format to the second node. The method then responds to the first communications command from the second node to the first node. The method then initiates from the second node a second communications command in the peer protocol format to the first node. The method then responds to the second communications command from the first node to the second node. The method then identifies the second device on the second node as a peer device within the first device on the first node, and the method identifies the first device on the first node as a peer device within the second device on the second node. The method then sets the first node to an active status, and sets the second node to a

passive status. In one embodiment, the peer protocol format can be a programmable logic controller (PLC) format, the peer device can be a programmable logic controller (PLC) device, the peer protocol format can be Modbus, and the standard communications network can be Ethernet.

The present invention is also a device scanner 110 for a first device 200 located on a first node of a standard communications network 10, for scanning devices on the standard communications network 10, and for identifying a second device 210 on a second node of the standard communications network 10. The device scanner 110 has an initiator for initiating a first communications command in a peer protocol format to the second node, a receptor for receiving from the second node a second communications command in the peer protocol format, in response to the first communications command, and an identifier for identifying the second device 210 on the second node as a peer device. The device scanner 110 can have the scan table 120 of prior embodiments built into the device scanner 110 or as a separate portion of the system for storing parameters relating to the devices. As in prior embodiments, the scanner 110 uses one or more of the parameters for scanning the devices.

In one particular embodiment, if communication has been initiated by a device which is in the scan table 120, that is, a device which this NOE 100 has been configured to also communicate with, then the peer determination test is performed. As indicated above , the peer determination includes at least the following aspects: 1) Does the remote device understand the read/write register command? 2) Does the communication from the remote device match the characteristics of this scan table 120. That is, does the write length match the read length in the scan table, and does the read length match the write length in the scan table 120. If (1) and (2) are met then the device is flagged as a "peer." There are two peer types: peer active and peer passive. The active peer takes over the task of initiating the scanning, while the passive peer only keeps track of the health of the active peer.

More particularly, as a part of its operation, the I/O scanner 110 determines during initialization whether an I/O device listed in the I/O scan table 120 is a peer device, which will actively initiate transfers, or a simple slave device, in which case the I/O Scanner must issue MB reads or writes to get or receive data. To determine peer status, the I/O scanner 110 issues a MB read/write request (which is FC 23) to the I/O device. If the /IO device responds with an exception indicating that it does not support the read/write request, then the I/O device is assumed to be a simple device, and therefore this I/O scanner 110 must initiate all requests for input data. If the device responds positively then it may or may not be a peer. The next qualifying event to declare the I/O device a peer is the arrival of a read/write request from the peer I/O device. If the read/write request is received and the input and output length match the configuration in this I/O scanner's scan table 120 then the device is declared a peer.

As briefly mentioned above, when a device is declared a peer, then the I/O devices IP addresses are used to decide which I/O device will be "active" and which I/O device will be "passive." By convention, the I/O device with the lower IP address will become active (lower = active) and the device with the higher IP address will become passive (higher = passive) in the I/O scanning process. The active device initiates the read, write, or read/write request to the passive I/O device. The passive device accepts/provides data in response to the active I/O device's requests. Figure 4 shows the timeline for the peer determination:

T0 - Node #1 initiates a Read/Write MSTR to Node #2.

T1 - Node #2 responds to Node #1's request.

T2 - Node #2 initiates a Read/Write MSTR to Node #1.

T3 - Node #1 responds to Node #2's request.

T4 - Node #1 declares Node #2 a Peer.

Node #2 then declares Node #1 a Peer.

Node #1 then becomes active and Node #2 becomes peer passive.

The following provides additional detail of one embodiment of the scanner 110 and scan table 120 of the present invention. The I/O scan table 120 allows up to a maximum of a 128 input devices and a maximum of 128 output devices. The I/O scan table allows up to a 100 words of data to be sent to or from a device in the I/O scan table 120. The format of the I/O scan table 120 is as follows:

Entry Name	Data Type	Description	Default Value
IODevType	Uint8	Device Type: Indicates the type of IO device	SLAVE_INPUT
Ipaddr	Uint32	IP address of IO Device	Derived from MB+ address
MbpAddr	Uint8	MB+ Address of IO Device	Value from Peer Cop Table
LastRespRecv	Uint8	Flag indicating whether the IO device responded to the last MB message	Reset when MB message sent. Set when response received
HealthTimeOutValue	Uint16	Indicates the amount of time t wait before declaring an IO device unhealthy	Value from Peer Cop Table
Status	Uint8	Indicates the status of the IO Device: UNCONNECTED, CONNECTED, TIMEDOUT.	UNCONNECTED

HealthTimer	Uint16	Used internally for current value of Health Timer	Updated once per scan. LSB is 16.67 mSec (from KC_TICKS2SEC).
InputLocalRefNum	Uint16	Local state RAM reference	Value from Peer Cop Table
InputRemoteRefNum	Uint16	Remote state RAM reference	0x00
InputLength	Uint16	Length of Input Data	Value from Peer Cop Table
HoldLastValue	Uint8	Indicates whether to hold the last value, or reset value to 0 when IO device is declared unhealthy	Value from Peer Cop Table
NewInputDataAvailable	Uint8	Indicates whether new input data has been received	Set when new data received, cleared when data is given to BP Driver.
IOScanDataInTblIndex	Uint16	Index into the IO Scan Input Data Table	Derived at initialization time
OutputLocalRefNum	Uint16	Local state RAM reference	Value from Peer Cop Table
OutputRemoteRefNum	Uint16	Remote state RAM reference	0x00
Output Length	Uint16	Length of Output Data	Value from Peer Cop Table
IOScanDataOutTblIndex	Uint16	Index into the IO Scan Output Data Table	Derived at initialization time

During the IOScanReadCfg state, the I/O scanner 110 initializes the connection and transaction arrays, then transitions to the IOScanRunning state. During the IOScanRunning state, the I/O scanner is issuing write/read registers(FC 23) to I/O devices in the scan table 120 that have both inputs and outputs, write4x register, and read4x register commands to I/O devices the scan table 120. The data received back from the read4x responses is sent to the backplane driver to update the controller memory.

The following describes with more particularity the peer active processing of one embodiment of the present invention. In the I/O peer case, the initial FC 23 from the peer, of the peer determination will be received by the server or server task 190, as well as, the FC 23s from peer active devices. The server 190 will determine whether the device associated with an incoming FC 23 is in the I/O scanner's I/O scan table 120. If it is, the server 190 will take control of the I/O scan table 120 by using the I/O scan table semaphore. Once the server 190 has control of the I/O scan table 120, the server 190 will determine whether the device associated with the incoming FC 23 is already flagged a peer, or is listed as a slave.

If the device is listed as a slave in the I/O scan table, then this FC 23 is part of the peer determination The server 190 then posts a IOSCAN_PEER_DETERMINATION_EVENT to the I/O scanner 110. Upon receipt of the IOSCAN_PEER_DETERMINATION_EVENT the I/O scanner 110 will update the status of the device to either peer active or peer passive, based on the IP addresses. If the device is listed as peer active, then this FC 23 is the I/O scan data associated with this peer. The server 190 will take control of the I/O scan table 120 by claiming the semaphore. The server 190 will write the new data into the I/O scan table 120 and update the NewInputDataAvailable flag. The server 190 will then release the I/O scan table 120 semaphore. If the output data length is non-zero, the server 190 will take control of the I/O scan table 120, and write the output data in the response to the FC

23 to the peer. The I/O scan table 120 can be broken down into two or more groups in order have the server 190 and/or the scanner 110 take control of only one or more groups therein and allow for access of the other groups by the other task. In peer passive, the response to the peer data exchange is received directly by the IO scanner 110, and is treated the same as the slave case.

The following describes with more particularity the input device scan operation of one embodiment of the present invention. While the I/O scanner 110 is in the IOScanRunning state, the BP driver sends an event to the I/O scanner 110 at each End of Scan (EOS). The transmission of the request for data for each input in the Scan Table is performed at the EOS. The I/O scanner 110 sends a FC 23 or FC 3 for each slave input, and peer active input in the scan table 120, if the device has responded to the previous request for data. The I/O scanner 110 clears the I/O data received flag for each request that is sent. The input data that is received as a result of the FC 23 or FC 3 commands that were transmitted will cause a IOSCAN TCIP_EVENT to be caused. This event is generated by the TCPSignalHandler function. The IOSCAN_TCIP_EVENT is handled by the I/O scanner 110 by determining which connection the response was received over, and setting the I/O data received flag for that device. The health timer is reset, and the health bit is set for the device.

The following describes with more particularity the ouput device scan operation of one embodiment of the present invention. Following the input device scan operation, the output devices are scanned. The output data is sent to each slave output, and peer active output in the scan list if the device has acknowledged the previous output data. When a device on the output scan list acknowledges the receipt of the output data, the health timer for the device is reset, and the health bit is set. To maintain consistency for all output data sent each scan, the I/O scanner 110 double buffers the output data before the starting the output of the data for each scan. Thus, if the BP driver updated part of the output data, while the I/O scanner 110 was in the

middle of a transmission, the current set of data being output would not be affected.

The following describes with more particularity the I/O device health information operation of one embodiment of the present invention. The health timer for each device is initially set to the health time-out value in the I/O scan table 120. The real time operating system timer 150 capability is used to maintain the health timers for each device in the I/O Scan table 120. The RTOS 150 system timer is configured to generate an event to the I/O scanner 110 every 16.67mSec. This is accomplished via the tm_evevery(unsigned long ticks, unsigned long events, unsigned long *tmid) function call(see PSOS documentation for more details). This configures a timer that will automatically reload. Every time the timer expires, an event is generated to the I/O scanner 110 indicating that the health timers need to be decremented. Following the update, any device whose health timer has expired, are flagged as bad by resetting the health bit for the device. Any time data is received from an input device or an acknowledge is received from an output device, the health timer for that device is set to the initial health time-out value. In order to use the I/O device health information with the current MSTR to get "peer cop" health, a 128 bit array will be used to provide health information for 128 devices. Otherwise, health will be provided for the first 64 devices. The I/O scan table 120 will be sorted for ascending IP addresses. Each bit in the 128 bit array indicates the health of one of the I/O devices. The I/O scanner 110 is capable of 1,000 transactions per second.

Th following provides additional information on the relationship between the I/O scanner 110, the server 190, and the backplane driver 180. In one embodiment of the present invention, three components are used to perform I/O scanning: the backplane driver 180, the server 190, and the I/O scanner 110. A double buffer scan table and a double buffer output table is used. The double buffer output table is used for outgoing write requests. An input table is also used

for outgoing read requests. A health array is also used. When the configuration changes, the backplane driver 190 copies the configuration from the PLC 50 into one configuration table while the client and server tasks are using another configuration table. When the backplane driver 190 has completed copying the configuration, it signals the client I/O scanner 110 task by setting the event flag. The client task 160 then swaps the tables, and the new table will also be used by the server 190. It is the client task 160 that determines which tables (configuration and output) are used by the server 190 and backplane driver 180. It does this with one variable for the configuration table and another variable for the output table. The server 190 and backplane driver 180 reads the appropriate variable and determines the table to use, and the client task 160 sets the appropriate variable. The server 190 examines if the read, write or read-write request from the remote node has a corresponding entry in the scan table 120. If not, the request is processed in the normal manner by passing the request to the backplane driver 180. If there is a corresponding entry, the server 190 processes it using the input and output tables. To save bandwidth, the server 190 compares its IP address with the remote node. If its IP address is greater than the remote node, it goes into passive mode, and changes the state in the scan table 120. At a later time, when the client task 160 notices that its in a passive state, it no longer sends modbus requests to the remote node. Thus, bandwidth is saved by sending less messages.

Referring back to embodiment in Figures 1 and 2, the I/O scanner 110 handles the cyclic communication, the server 190 communicates directly with a dual port RAM and an ASIC which is directly connected to a bus that runs accross the backplane 70 which in turn connects to the PLC 50. The backplane driver 180 handles communication to/from the controller. As discussed above, the PLC 50 has configuration tables which stores, at least, the number of devices to be scanned, whether a device is an input, and output, or and input and output device.

If a device is an input, the table stores where in the controller's memory to store the data. If the device is an output, the table stores where in the memory to retrieve the data. If the device is both an input and output device, the table stores both. The configuration table also stores the length (in bytes) of the input and output data. As described in detail above, the configuration table also includes a health timeout parameter for each device. If there is no response to a read/write to an I/O device within the timeout parameter amount, the device can be "flagged" as unhealthy in the NOE 100. The NOE 100 keeps track of the dynamic health status.

In this embodiment, the configuration table is scanned by the NOE 100, and then the NOE 100 operates according to those parameters. The controller (PLC) 50, 52 runs in a cyclic manner, and it handles the updating of the inputs and outputs once per scan. At the end of each scan, the controller gives an indication tp the NOE 100 that the controller is at the end of a scan, and the NOE 100 then takes a snapshot of all of the available outputs and copies the outputs into its local memory. The NOE 100 then generates messages to update all of the output devices that are in the I/O scan table. The NOE 100 also takes the current values for all of the input devices (after sending read messages for all of the input devices) and the NOE 100 updates the controller 50 with all of the new input data.

In Figure 2, there is an indication of a "semaphore controlled" I/O scan table 120. This term was briefly mentioned above. The scan table 120 needs to be available to both the I/O scanner 110 and the server 190 because the server 190 needs to be able to talk to a "peer" device that is generating messages. Those messages will go to the server 190 first (before the I/O scanner 110). Generally, the server 190 receives requests on the Modbus port 502 from other controllers 50, 52 and NOEs 100. Continuing, in order to prevent both the server 190 and the I/O scanner 110 from accessing the I/O scan table 120 at the same time, a control

mechanism is provided to control "ownership" of the table at any one time by either the I/O scanner 110 or the server 190. If this is not done, problems with consistency of the data and its use may arise.

The client task 160 is a portion of the NOE 100 that handles "client" tasks. A described above, users of the controllers will create programs to run on the controllers. The programs can be written to send Modbus messages over TCP. The controller 50, 52 will send a message to the backplane 70 (type of message will be handles by the client task 160) and the client task 160 will send the message onto the TCP/IP over Ethernet network. A message will then come back from the TCP/IP over Ethernet network, the client task will handle the return message, and send it onto the controller 50, 52 for use by the program running on the controller 50, 52. The Client Connection Function Library 128 is a library of functions which are used to format and unformat messages to/from the format on the backplane 70 handled by the backplane driver 180, as well as the format of the TCP/IP stack 142, including the handling of the Modbus format.

A detailed flow diagram (chart) of one embodiment of the I/O scanner 110 of the present invention is shown in Figures 5A through 5G, as one of ordinary skill would understand. This embodiment does not need to use an RTOS 150 for its operation, and thus includes some additional routines for handling. The program of this detailed flow diagram can be loaded into flash program firmware memory on the NOE.

While the specific embodiments have been illustrated and described, numerous modifications come to mind without significantly departing from the spirit of the invention and the scope of protection is only limited by the scope of the accompanying Claims.

CLAIMS

WE CLAIM:

1. An apparatus for communication with at least one device which resides on a standard communications network using a standard communications protocol, comprising:

a scanner for scanning the device;

a device scan table for storing data relating to the device, the scanner using the data in the device scan table relating to the device to scan the device; and,

a standard communications interface for interfacing between the scanner and the device on the standard communications network using the standard communication protocol.

2. The apparatus of claim 1 wherein the device scan table comprises:

a first scan parameter indicating the number of devices to be scanned by the device scanner;

a second scan parameter indicating a device type;

a third scan parameter indicating where to store and retrieve data for the devices; and,

a fourth scan parameter indicating the length of the stored data and the retrieved data.

3. The apparatus of claim 1, wherein the standard communications

network is an Ethernet network.

4. The apparatus of claim 1, wherein the standard communications protocol is TCP.

5. The apparatus of claim 1, wherein a programmable logic controller (PLC) communication protocol is communicated over the standard communications protocol.

6. The apparatus of claim 1, wherein the standard communications network provides communication between the device scanner and the devices.

7. An apparatus for monitoring and controlling input and output devices which reside on a standard communications network using a standard communications protocol, comprising:

a scanner for scanning the input and output devices;

an input/output (I/O) scan table for storing input and output data relating to the input and output devices; and,

a standard communications interface for interfacing between the I/O scanner and the standard communications network using the standard communication protocol.

8. The apparatus of claim 7 wherein the I/O scan table comprises:

a first scan parameter indicating the number of devices to be scanned by the I/O scanner;

a second scan parameter indicating whether each device is an input device, an output device, or an input and output device;

a third scan parameter indicating where to store data from the input devices;

a fourth scan parameter indicating where to retrieve data for the output devices; and,

a fifth scan parameter indicating the length of the stored data and the retrieved data, for the input and output devices, respectively.

9. The apparatus of claim 7, wherein the standard communications network is an Ethernet network.

10. The apparatus of claim 7, wherein the standard communications protocol is TCP.

11. The apparatus of claim 7, wherein a programmable logic controller (PLC) communication protocol is communicated over the standard communications protocol.

12. The apparatus of claim 7, wherein the standard communications network provides communication between the I/O scanner and the input and output devices.

13. A control system for monitoring input devices and for controlling output devices which reside on a standard communications network using a standard communication protocol, the control system comprising:

a programmable logic controller (PLC) having a microprocessor and a PLC memory having a configuration table, the PLC being adapted to communicate with at least one or more local input and output devices, and being adapted to communicate with at least one or more remote input and output devices, the configuration table storing parameters for the input and output devices, the parameters including one or more of the following parameters: a first configuration parameter indicating the number of devices to be scanned, a second configuration parameter indicating whether each device is an input device, an output device, or an input and output device, a third configuration parameter indicating where in the PLC memory to store data from the input devices, a fourth configuration parameter indicating where in the PLC memory to retrieve data for the output devices, and a fifth configuration parameter indicating the length of the stored data and the retrieved data, for the input and output devices, respectively;

a server module adapted for communication with the PLC, adapted for communication with local input and output devices, and adapted for communication with the standard communication network and the remote input and output devices, the server module having: a scanner for scanning the input and output devices, an I/O scan table for storing real time information for the input and output devices, a standard communication protocol interface and a standard communication network driver for interfacing between the I/O scanner and the

standard communication network using the standard communication protocol, a real time operating system for operating the server module, wherein the I/O scan table comprises:

a first scan parameter indicating the number of devices to be scanned by the I/O scanner;

a second scan parameter indicating whether each device is an input device, an output device, or an input and output device;

a third scan parameter indicating where in the PLC memory to store data from the input devices;

a fourth scan parameter indicating where in the PLC memory to retrieve data for the output devices; and,

a fifth scan parameter indicating the length of the stored data and the retrieved data, for the input and output devices, respectively.

14. A method for identifying a second device on a second node of a standard communications network from a first device located on a first node of the standard communications network, comprising the steps of:

initiating from the first node a first communications command in a peer protocol format to the second node;

initiating from the second node a second communications command in the peer protocol format to the first node;

identifying the second device on the second node as a peer device within the first device on the first node; and,

identifying the first device on the first node as a peer device within the second device on the second node.

15. The method of claim 14, wherein the peer protocol format is a programmable logic controller (PLC) format.

16. The method of claim 14, wherein the peer device is a programmable logic controller (PLC) device.

17. The method of claim 14, wherein the peer protocol format is Modbus.

18. The method of claim 14, wherein the standard communications network is Ethernet.

19. The method of claim 14, further comprising the step of:

after initiating from the first node a first communications command in a peer protocol format to the second node, responding to the first communications command from the second node to the first node.

20. The method of claim 14, further comprising the step of:

after initiating from the second node the second communications command in the peer protocol format to the first node, responding to the second communications command from the first node to the second node.

21. The method of claim 14, further comprising the steps of:

setting the first node to an active status; and,

setting the second node to a passive status.

22. A device scanner for a first device located on a first node of a standard communications network, for scanning devices on the standard communications network, and for identifying a second device on a second node of the standard communications network, comprising:

an initiator for initiating a first communications command in a peer protocol format to the second node;

a receptor for receiving from the second node a second communications command in the peer protocol format, in response to the first communications command; and,

an identifier for identifying the second device on the second node as a peer device.

23. The device scanner of claim 22 further comprising a scan table for storing parameters relating to the devices, the scanner using one or more of the parameters for scanning the devices.

24. The device scanner of claim 22 wherein the second device identifies the first device on the first node as a peer device.

25. The device scanner of claim 22, wherein the peer protocol format is a programmable logic controller (PLC) format.

26. The device scanner of claim 22, wherein the peer device is a programmable logic controller (PLC) device.

27. The device scanner of claim 22, wherein the peer protocol format is Modbus.

28. The device scanner of claim 22, wherein the standard communications network is Ethernet.

29. The device scanner of claim 22, wherein after the device scanner initiates the first communications command, the second node responds to the first communications command to the first node.

30. The device scanner of claim 22, wherein the after the second communications command is received by the first node, the device scanner responds to the second communications command to the second node.

31. The device scanner of claim 22, wherein after the second device on the second node is identified as a peer device, the first node is set to an active status, and the second node is set to a passive status.

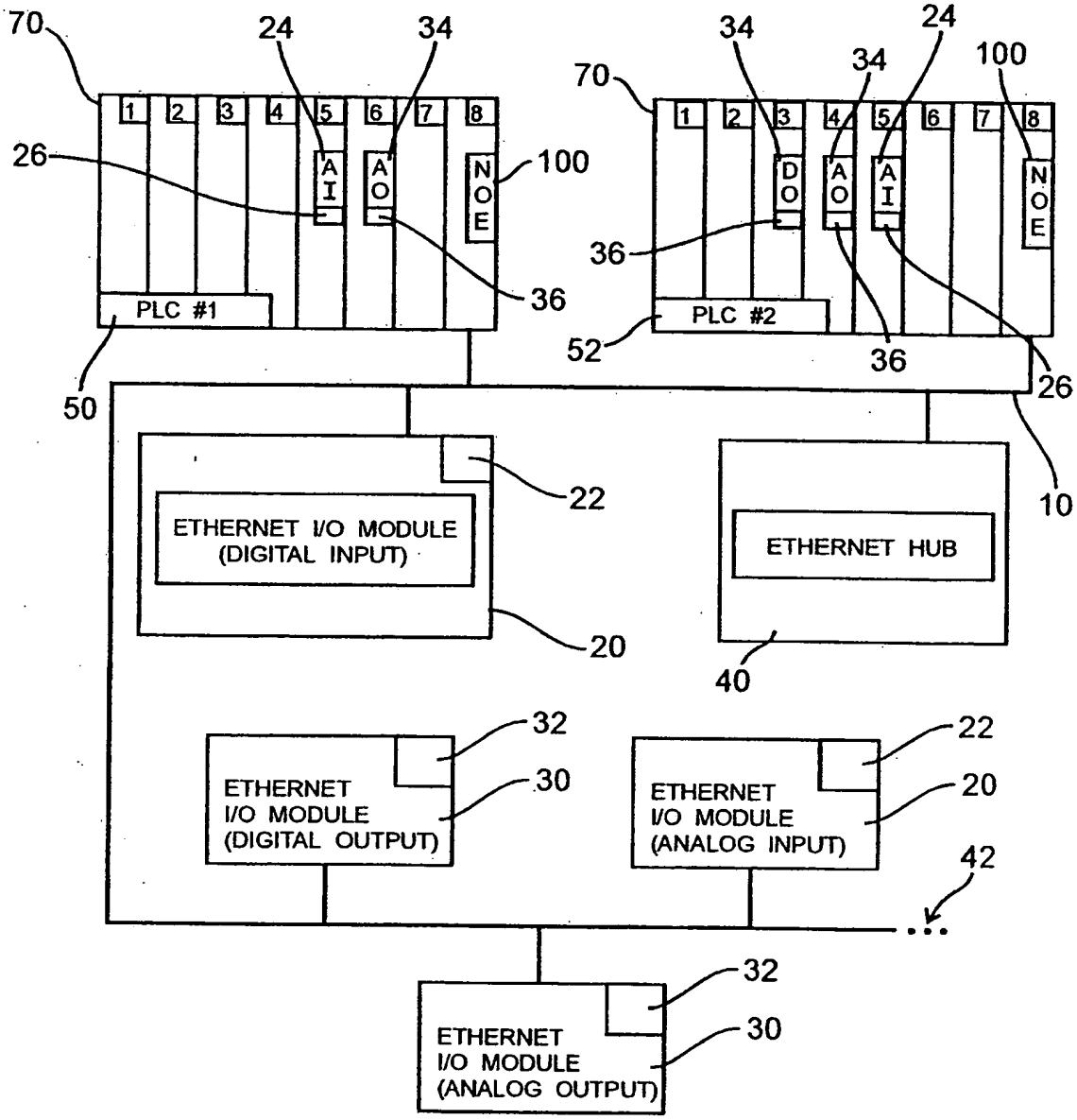

70
24
34
34
24
34
100
70
1
2
3
4
5
6
7
8
AI
AO
NOE
DO
26
100
36
PLC #1
PLC #2
52
36
26
50
22
10
ETHERNET I/O MODULE
(DIGITAL INPUT)
ETHERNET HUB
20
40
32
22
30
20
ETHERNET
I/O MODULE
(DIGITAL OUTPUT)
ETHERNET
I/O MODULE
(ANALOG INPUT)
42
32
30
ETHERNET
I/O MODULE
(ANALOG OUTPUT)

Fig. 1

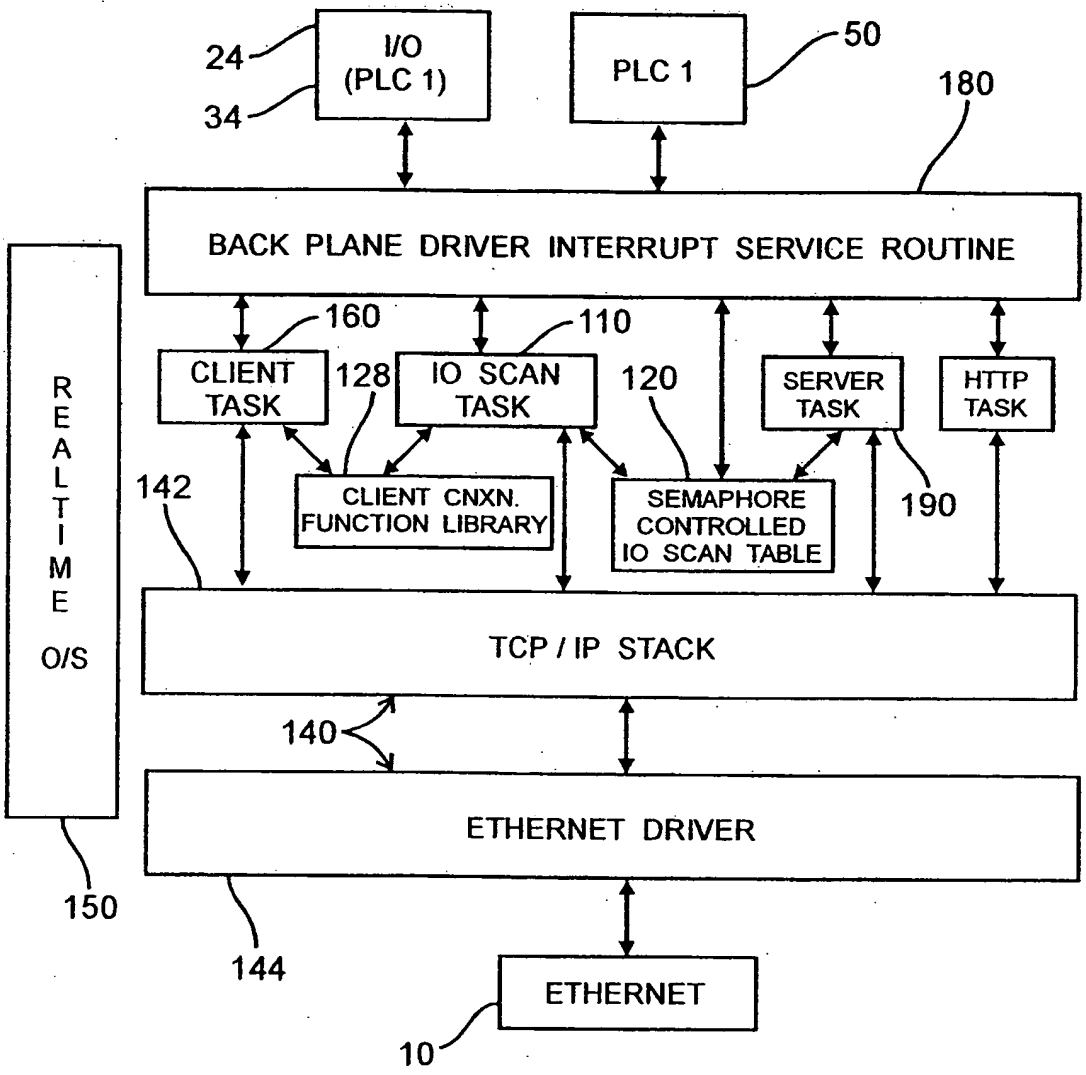

24
34
I/O
(PLC 1)
PLC 1
50
180
BACK PLANE DRIVER INTERRUPT SERVICE ROUTINE
REALTIME O/S
160
CLIENT TASK
128
IO SCAN TASK
110
120
SERVER TASK
HTTP TASK
142
CLIENT CNXN. FUNCTION LIBRARY
SEMAPHORE CONTROLLED IO SCAN TABLE
190
TCP / IP STACK
140
ETHERNET DRIVER
150
144
ETHERNET
10

Fig. 2

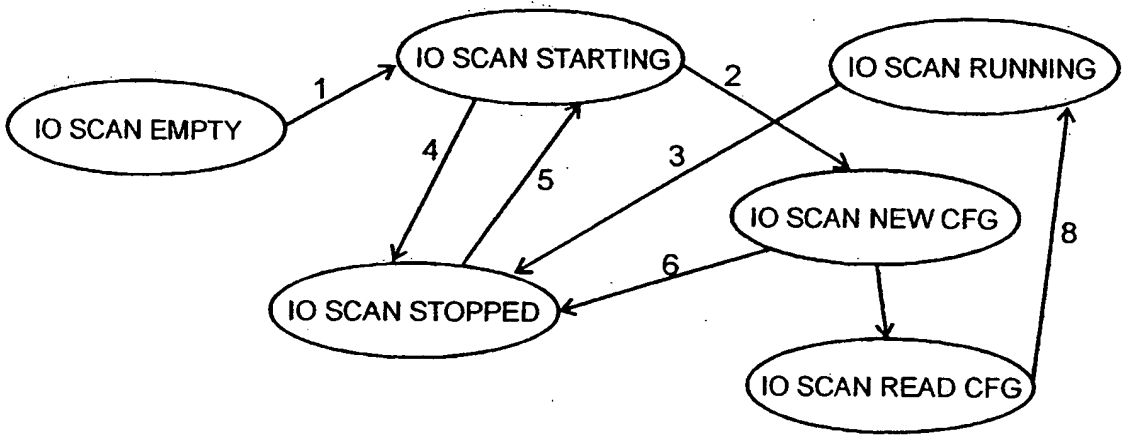

IO SCAN EMPTY
IO SCAN STARTING
IO SCAN RUNNING
IO SCAN NEW CFG
IO SCAN STOPPED
IO SCAN READ CFG
1
2
3
4
5
6
8

Fig. 3

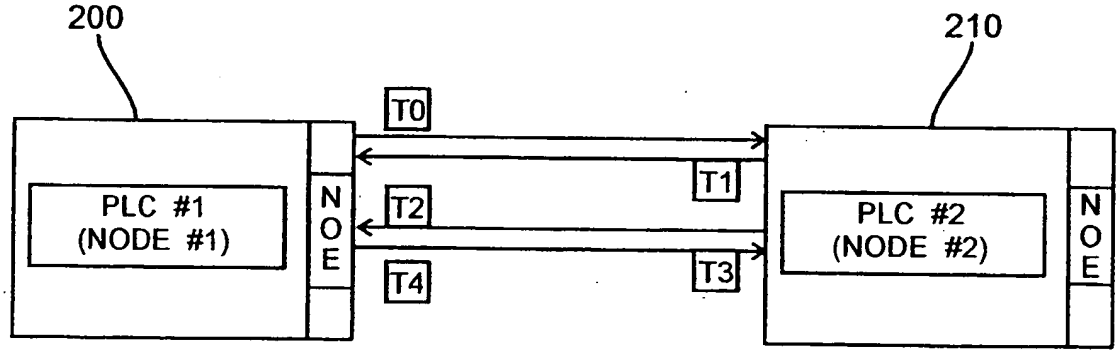

200
210
T0
T1
T2
T3
T4
PLC #1
(NODE #1)
NOE
PLC #2
(NODE #2)
NOE

Fig. 4

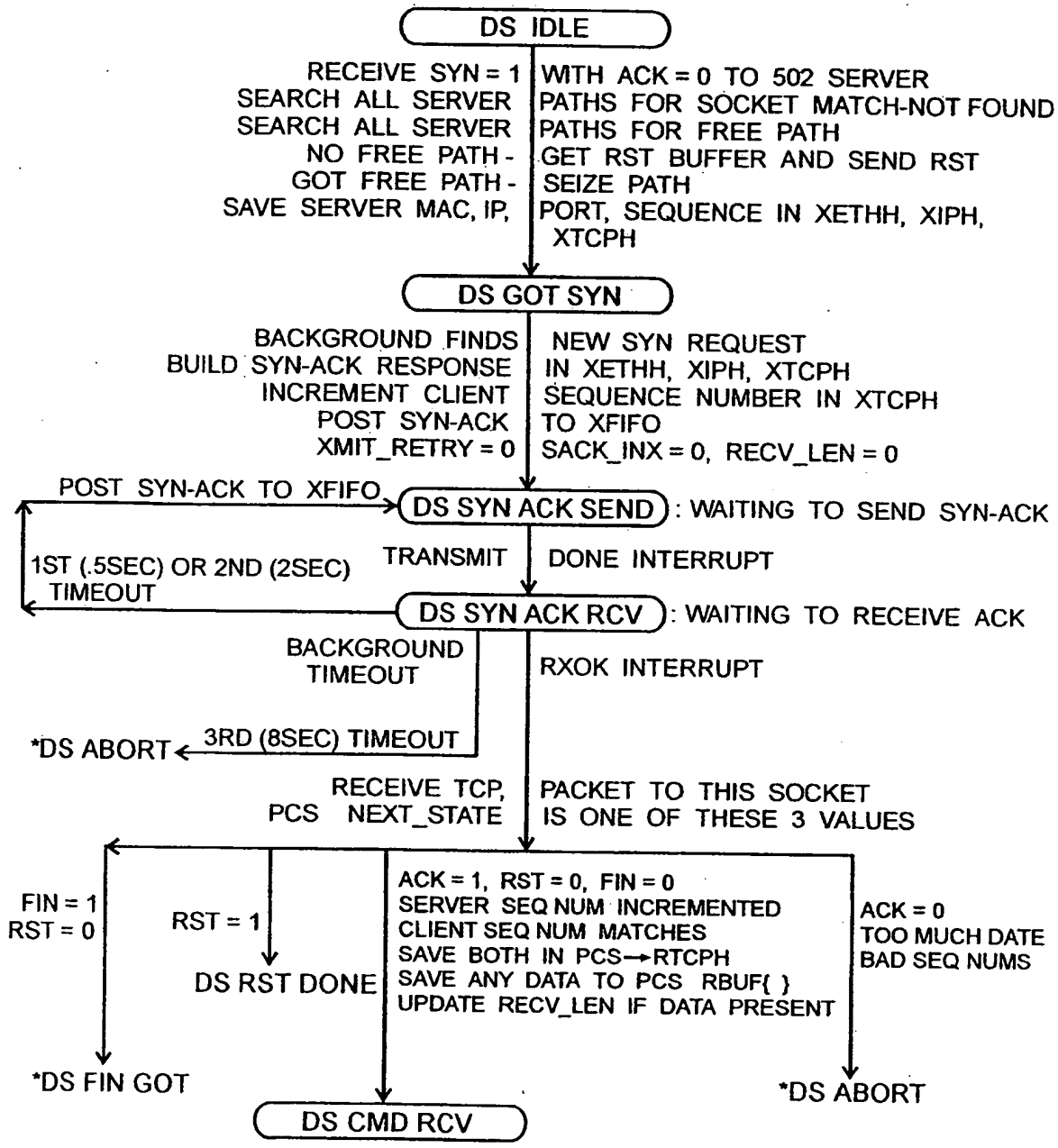

DS IDLE
RECEIVE SYN = 1
WITH ACK = 0 TO 502 SERVER
SEARCH ALL SERVER
PATHS FOR SOCKET MATCH-NOT FOUND
SEARCH ALL SERVER
PATHS FOR FREE PATH
NO FREE PATH -
GET RST BUFFER AND SEND RST
GOT FREE PATH -
SEIZE PATH
SAVE SERVER MAC, IP,
PORT, SEQUENCE IN XETHH, XIPH, XTCPH
DS GOT SYN
BACKGROUND FINDS
NEW SYN REQUEST
BUILD SYN-ACK RESPONSE
IN XETHH, XIPH, XTCPH
INCREMENT CLIENT
SEQUENCE NUMBER IN XTCPH
POST SYN-ACK
TO XFIFO
XMIT_RETRY = 0
SACK_INX = 0, RECV_LEN = 0
POST SYN-ACK TO XFIFO
DS SYN ACK SEND
: WAITING TO SEND SYN-ACK
TRANSMIT
DONE INTERRUPT
1ST (.5SEC) OR 2ND (2SEC) TIMEOUT
DS SYN ACK RCV
: WAITING TO RECEIVE ACK
BACKGROUND TIMEOUT
RXOK INTERRUPT
*DS ABORT
3RD (8SEC) TIMEOUT
RECEIVE TCP, PCS NEXT_STATE
PACKET TO THIS SOCKET IS ONE OF THESE 3 VALUES
FIN = 1
RST = 0
RST = 1
ACK = 1, RST = 0, FIN = 0
SERVER SEQ NUM INCREMENTED
CLIENT SEQ NUM MATCHES
SAVE BOTH IN PCS→RTCPH
SAVE ANY DATA TO PCS RBUF{ }
UPDATE RECV_LEN IF DATA PRESENT
ACK = 0
TOO MUCH DATE
BAD SEQ NUMS
DS RST DONE
*DS FIN GOT
*DS ABORT
DS CMD RCV

Fig. 5A

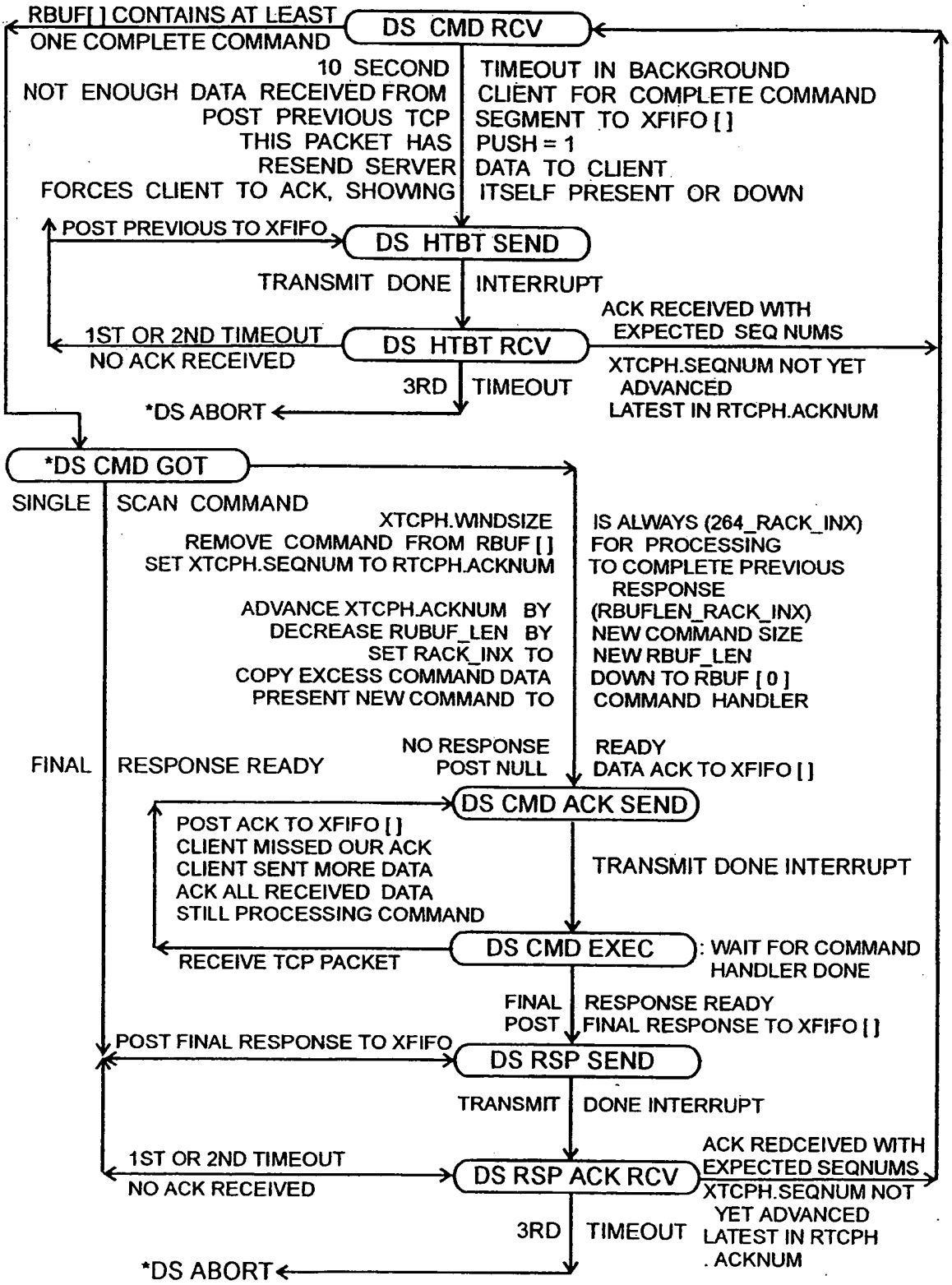

RBUF[ ] CONTAINS AT LEAST ONE COMPLETE COMMAND
DS CMD RCV
10 SECOND NOT ENOUGH DATA RECEIVED FROM POST PREVIOUS TCP THIS PACKET HAS RESEND SERVER FORCES CLIENT TO ACK, SHOWING
TIMEOUT IN BACKGROUND CLIENT FOR COMPLETE COMMAND SEGMENT TO XFIFO [ ] PUSH = 1 DATA TO CLIENT ITSELF PRESENT OR DOWN
POST PREVIOUS TO XFIFO
DS HTBT SEND
TRANSMIT DONE
INTERRUPT
1ST OR 2ND TIMEOUT NO ACK RECEIVED
DS HTBT RCV
ACK RECEIVED WITH EXPECTED SEQ NUMS
XTCPH.SEQNUM NOT YET ADVANCED LATEST IN RTCPH.ACKNUM
3RD TIMEOUT
*DS ABORT
*DS CMD GOT
SINGLE
SCAN COMMAND
XTCPH.WINDSIZE
REMOVE COMMAND FROM RBUF [ ]
SET XTCPH.SEQNUM TO RTCPH.ACKNUM
ADVANCE XTCPH.ACKNUM BY
DECREASE RUBUF_LEN BY
SET RACK_INX TO
COPY EXCESS COMMAND DATA
PRESENT NEW COMMAND TO
IS ALWAYS (264_RACK_INX)
FOR PROCESSING
TO COMPLETE PREVIOUS RESPONSE
(RBUFLEN_RACK_INX)
NEW COMMAND SIZE
NEW RBUF_LEN
DOWN TO RBUF [0]
COMMAND HANDLER
FINAL
RESPONSE READY
NO RESPONSE POST NULL
READY
DATA ACK TO XFIFO [ ]
DS CMD ACK SEND
POST ACK TO XFIFO [ ]
CLIENT MISSED OUR ACK
CLIENT SENT MORE DATA
ACK ALL RECEIVED DATA
STILL PROCESSING COMMAND
TRANSMIT DONE INTERRUPT
DS CMD EXEC
: WAIT FOR COMMAND HANDLER DONE
RECEIVE TCP PACKET
FINAL POST
RESPONSE READY
FINAL RESPONSE TO XFIFO [ ]
POST FINAL RESPONSE TO XFIFO
DS RSP SEND
TRANSMIT
DONE INTERRUPT
1ST OR 2ND TIMEOUT NO ACK RECEIVED
DS RSP ACK RCV
ACK REDCEIVED WITH EXPECTED SEQNUMS
XTCPH.SEQNUM NOT YET ADVANCED LATEST IN RTCPH . ACKNUM
3RD TIMEOUT
*DS ABORT

Fig. 5B

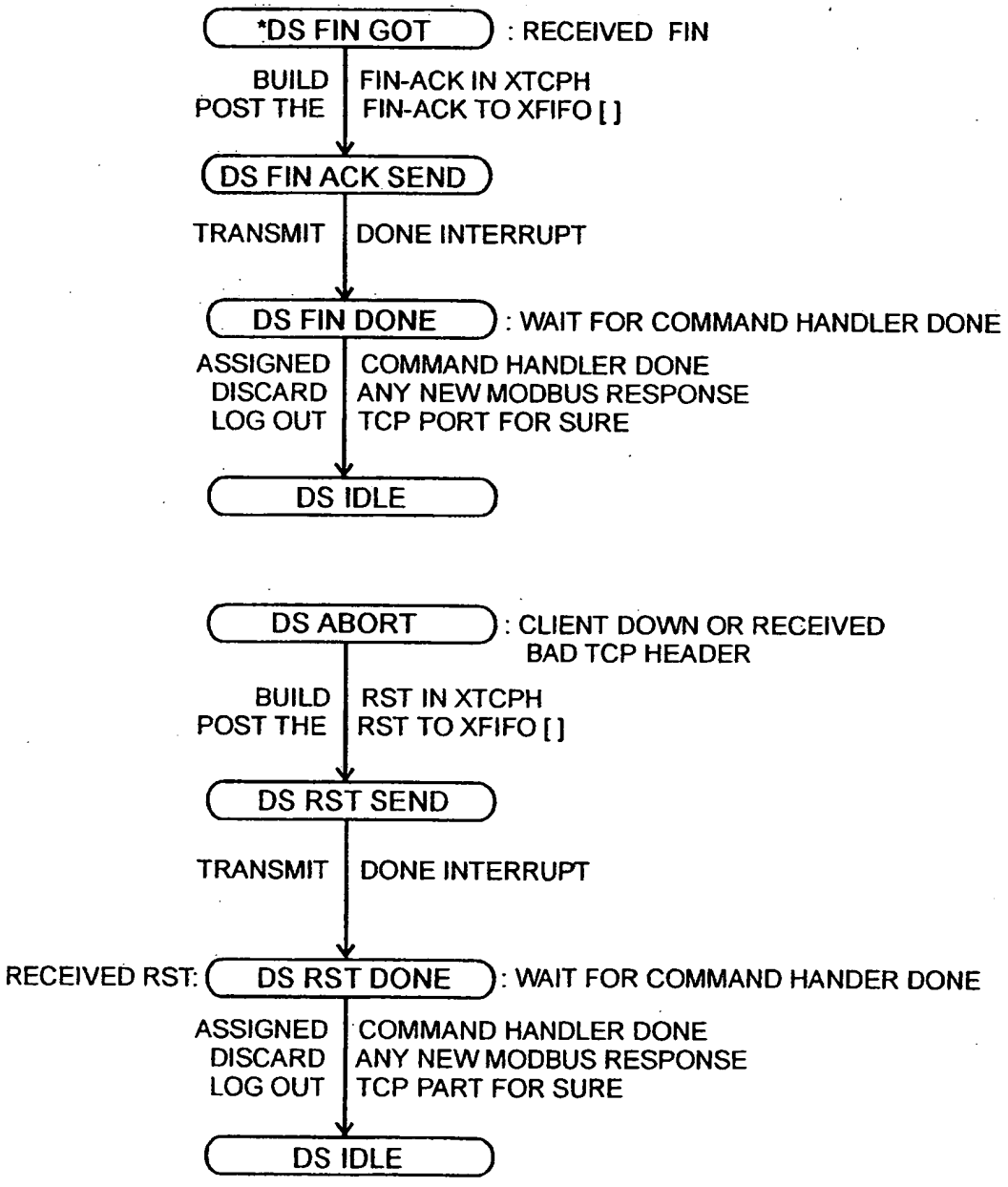

*DS FIN GOT
: RECEIVED FIN
BUILD
POST THE
FIN-ACK IN XTCPH
FIN-ACK TO XFIFO [ ]
DS FIN ACK SEND
TRANSMIT
DONE INTERRUPT
DS FIN DONE
: WAIT FOR COMMAND HANDLER DONE
ASSIGNED
DISCARD
LOG OUT
COMMAND HANDLER DONE
ANY NEW MODBUS RESPONSE
TCP PORT FOR SURE
DS IDLE
DS ABORT
: CLIENT DOWN OR RECEIVED
BAD TCP HEADER
BUILD
POST THE
RST IN XTCPH
RST TO XFIFO [ ]
DS RST SEND
TRANSMIT
DONE INTERRUPT
RECEIVED RST:
DS RST DONE
: WAIT FOR COMMAND HANDER DONE
ASSIGNED
DISCARD
LOG OUT
COMMAND HANDLER DONE
ANY NEW MODBUS RESPONSE
TCP PART FOR SURE
DS IDLE

Fig. 5C

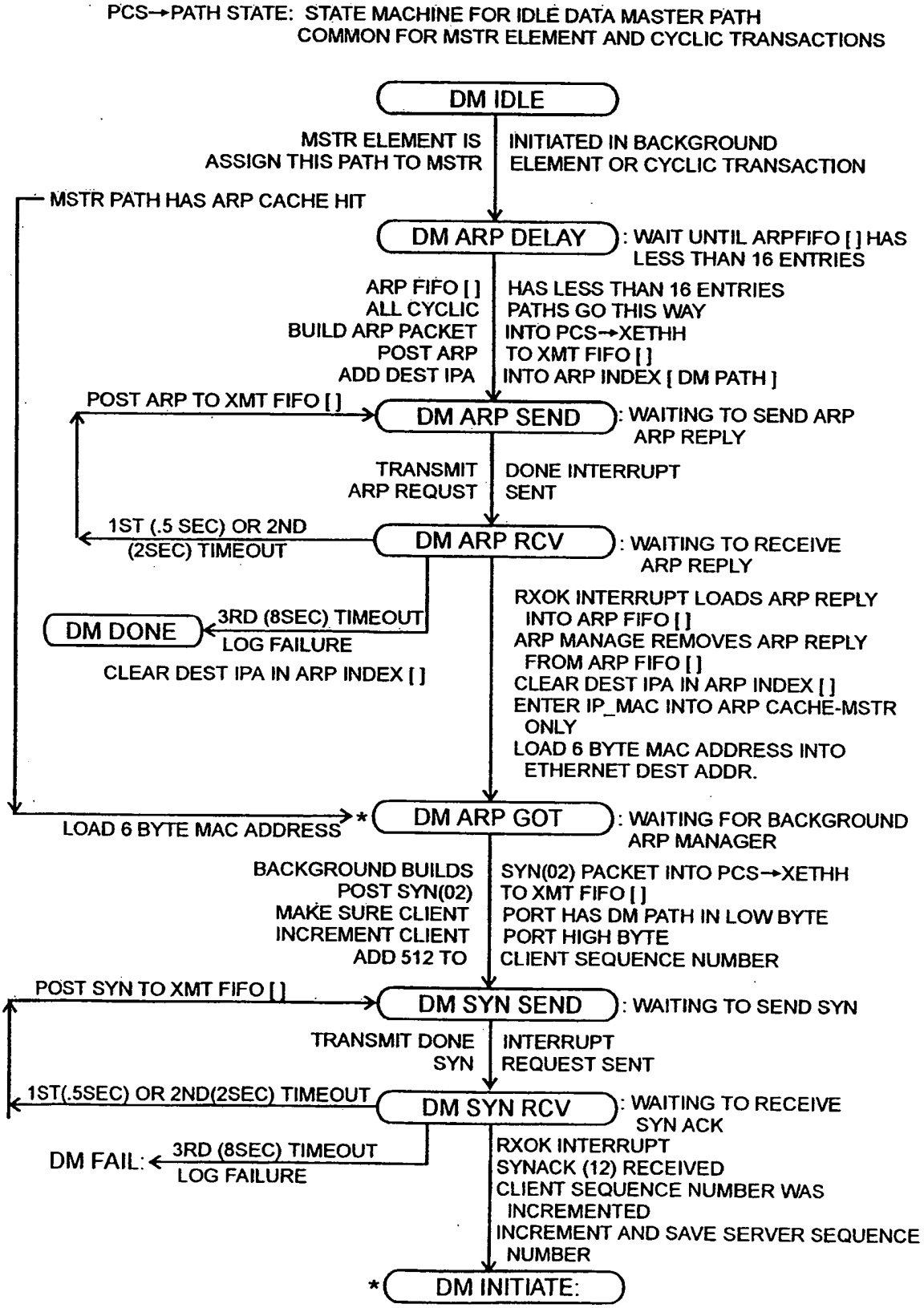

PCS→PATH STATE: STATE MACHINE FOR IDLE DATA MASTER PATH
COMMON FOR MSTR ELEMENT AND CYCLIC TRANSACTIONS
DM IDLE
MSTR ELEMENT IS ASSIGN THIS PATH TO MSTR
INITIATED IN BACKGROUND ELEMENT OR CYCLIC TRANSACTION
MSTR PATH HAS ARP CACHE HIT
DM ARP DELAY
: WAIT UNTIL ARPFIFO [ ] HAS LESS THAN 16 ENTRIES
ARP FIFO [ ] ALL CYCLIC BUILD ARP PACKET POST ARP ADD DEST IPA
HAS LESS THAN 16 ENTRIES PATHS GO THIS WAY INTO PCS→XETHH TO XMT FIFO [ ] INTO ARP INDEX [DM PATH]
POST ARP TO XMT FIFO [ ]
DM ARP SEND
: WAITING TO SEND ARP ARP REPLY
TRANSMIT ARP REQUST
DONE INTERRUPT SENT
1ST (.5 SEC) OR 2ND (2SEC) TIMEOUT
DM ARP RCV
: WAITING TO RECEIVE ARP REPLY
3RD (8SEC) TIMEOUT
LOG FAILURE
DM DONE
CLEAR DEST IPA IN ARP INDEX [ ]
RXOK INTERRUPT LOADS ARP REPLY INTO ARP FIFO [ ]
ARP MANAGE REMOVES ARP REPLY FROM ARP FIFO [ ]
CLEAR DEST IPA IN ARP INDEX [ ]
ENTER IP_MAC INTO ARP CACHE-MSTR ONLY
LOAD 6 BYTE MAC ADDRESS INTO ETHERNET DEST ADDR.
LOAD 6 BYTE MAC ADDRESS
*
DM ARP GOT
: WAITING FOR BACKGROUND ARP MANAGER
BACKGROUND BUILDS POST SYN(02) MAKE SURE CLIENT INCREMENT CLIENT ADD 512 TO
SYN(02) PACKET INTO PCS→XETHH TO XMT FIFO [ ] PORT HAS DM PATH IN LOW BYTE PORT HIGH BYTE CLIENT SEQUENCE NUMBER
POST SYN TO XMT FIFO [ ]
DM SYN SEND
: WAITING TO SEND SYN
TRANSMIT DONE SYN
INTERRUPT REQUEST SENT
1ST(.5SEC) OR 2ND(2SEC) TIMEOUT
DM SYN RCV
: WAITING TO RECEIVE SYN ACK
DM FAIL:
3RD (8SEC) TIMEOUT
LOG FAILURE
RXOK INTERRUPT
SYNACK (12) RECEIVED
CLIENT SEQUENCE NUMBER WAS INCREMENTED
INCREMENT AND SAVE SERVER SEQUENCE NUMBER
*
DM INITIATE:

Fig. 5D

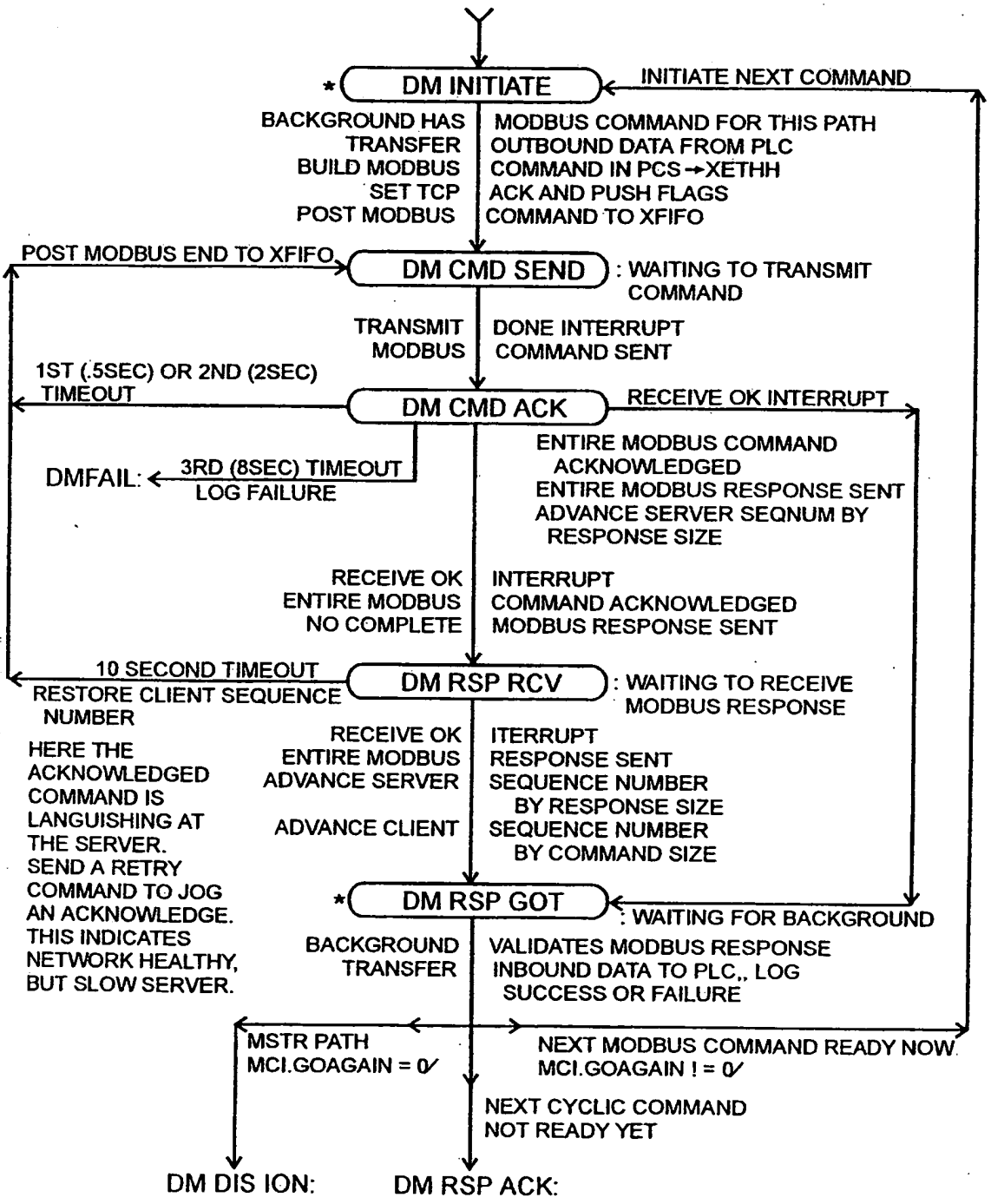

* DM INITIATE
INITIATE NEXT COMMAND
BACKGROUND HAS TRANSFER
BUILD MODBUS
SET TCP
POST MODBUS
MODBUS COMMAND FOR THIS PATH
OUTBOUND DATA FROM PLC
COMMAND IN PCS→XETHH
ACK AND PUSH FLAGS
COMMAND TO XFIFO
POST MODBUS END TO XFIFO
DM CMD SEND
: WAITING TO TRANSMIT COMMAND
TRANSMIT MODBUS
DONE INTERRUPT
COMMAND SENT
1ST (.5SEC) OR 2ND (2SEC) TIMEOUT
DM CMD ACK
RECEIVE OK INTERRUPT
DMFAIL:
3RD (8SEC) TIMEOUT
LOG FAILURE
ENTIRE MODBUS COMMAND ACKNOWLEDGED
ENTIRE MODBUS RESPONSE SENT
ADVANCE SERVER SEQNUM BY RESPONSE SIZE
RECEIVE OK
ENTIRE MODBUS
NO COMPLETE
INTERRUPT
COMMAND ACKNOWLEDGED
MODBUS RESPONSE SENT
10 SECOND TIMEOUT
RESTORE CLIENT SEQUENCE NUMBER
DM RSP RCV
: WAITING TO RECEIVE MODBUS RESPONSE
HERE THE ACKNOWLEDGED COMMAND IS LANGUISHING AT THE SERVER. SEND A RETRY COMMAND TO JOG AN ACKNOWLEDGE. THIS INDICATES NETWORK HEALTHY, BUT SLOW SERVER.
RECEIVE OK
ENTIRE MODBUS
ADVANCE SERVER
ITERRUPT
RESPONSE SENT
SEQUENCE NUMBER BY RESPONSE SIZE
ADVANCE CLIENT
SEQUENCE NUMBER BY COMMAND SIZE
* DM RSP GOT
: WAITING FOR BACKGROUND
BACKGROUND TRANSFER
VALIDATES MODBUS RESPONSE
INBOUND DATA TO PLC,, LOG SUCCESS OR FAILURE
MSTR PATH
MCI.GOAGAIN = 0
NEXT MODBUS COMMAND READY NOW
MCI.GOAGAIN ! = 0
NEXT CYCLIC COMMAND NOT READY YET
DM DIS ION:
DM RSP ACK:
* - RESIDES IN THIS STATE FOR Ø TIME
RCV INTERRUPT SETS NEXT STATE TO THESE VALUES

Fig. 5E

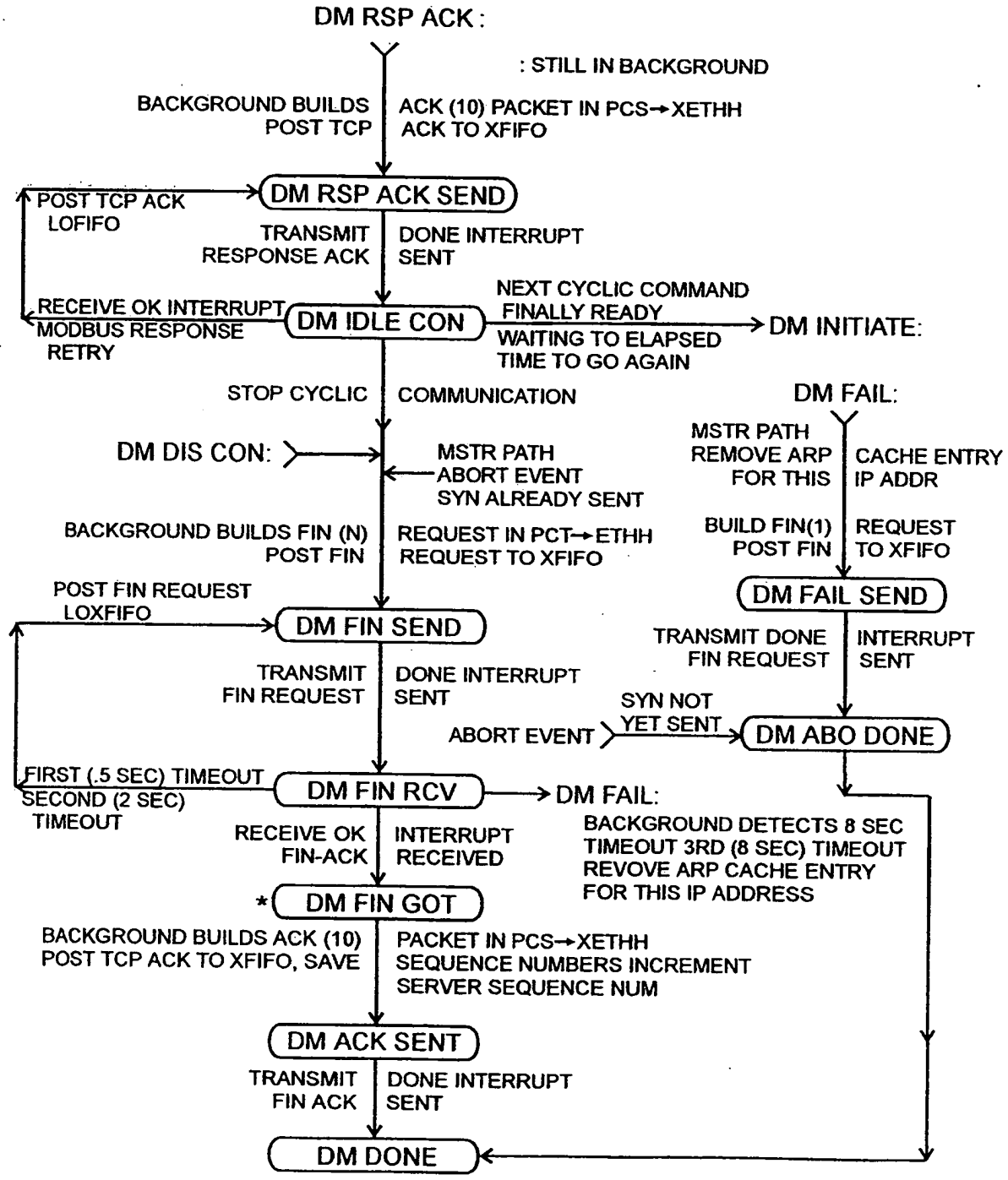

DM RSP ACK:
: STILL IN BACKGROUND
BACKGROUND BUILDS POST TCP
ACK (10) PACKET IN PCS→XETHH ACK TO XFIFO
DM RSP ACK SEND
POST TCP ACK LOFIFO
TRANSMIT RESPONSE ACK
DONE INTERRUPT SENT
RECEIVE OK INTERRUPT MODBUS RESPONSE RETRY
DM IDLE CON
NEXT CYCLIC COMMAND FINALLY READY
WAITING TO ELAPSED TIME TO GO AGAIN
DM INITIATE:
STOP CYCLIC
COMMUNICATION
DM FAIL:
DM DIS CON:
MSTR PATH ABORT EVENT SYN ALREADY SENT
MSTR PATH REMOVE ARP FOR THIS
CACHE ENTRY IP ADDR
BACKGROUND BUILDS FIN (N) POST FIN
REQUEST IN PCT→ETHH REQUEST TO XFIFO
BUILD FIN(1) POST FIN
REQUEST TO XFIFO
POST FIN REQUEST LOXFIFO
DM FIN SEND
DM FAIL SEND
TRANSMIT DONE FIN REQUEST
INTERRUPT SENT
TRANSMIT FIN REQUEST
DONE INTERRUPT SENT
SYN NOT YET SENT
ABORT EVENT
DM ABO DONE
FIRST (.5 SEC) TIMEOUT SECOND (2 SEC) TIMEOUT
DM FIN RCV
DM FAIL:
RECEIVE OK FIN-ACK
INTERRUPT RECEIVED
BACKGROUND DETECTS 8 SEC TIMEOUT 3RD (8 SEC) TIMEOUT REVOVE ARP CACHE ENTRY FOR THIS IP ADDRESS
* DM FIN GOT
BACKGROUND BUILDS ACK (10) POST TCP ACK TO XFIFO, SAVE
PACKET IN PCS→XETHH SEQUENCE NUMBERS INCREMENT SERVER SEQUENCE NUM
DM ACK SENT
TRANSMIT FIN ACK
DONE INTERRUPT SENT
DM DONE

Fig. 5F

PCS→XMIT_STATE

STATE MACHINE: ONCE A PATH HAS BEEN SUBMITTED INTO THE XFIFO [ ], IT CANNOT BE REMOVED. BUT THE BACKGROUND CAN ABORT THE PACKET, AS LONG AS IT IS NOT THE XMT BID OR XMT SEND STATE. THESE STATES SHOULD NOT PERSIST.

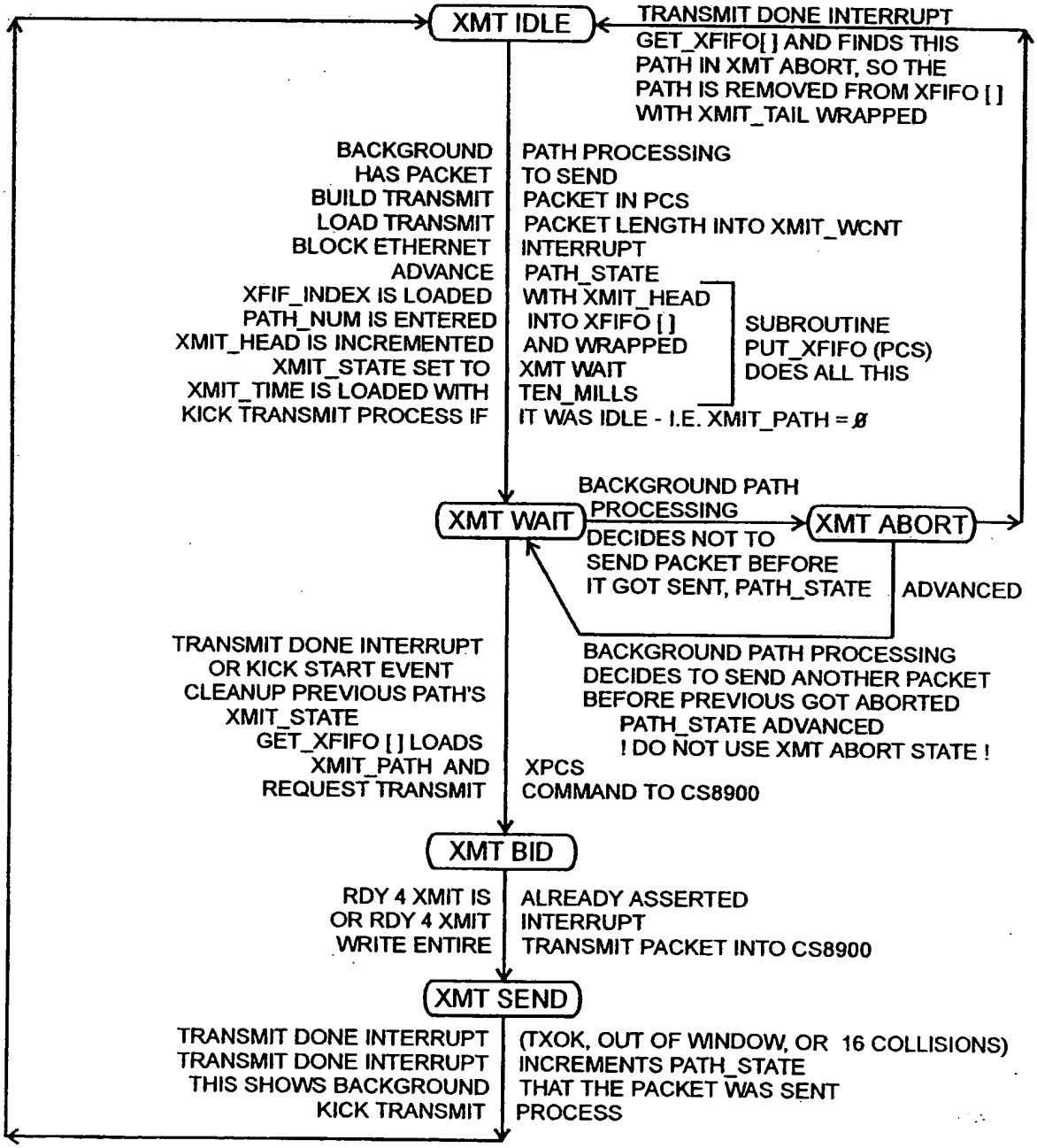


Fig. 5G

INTERNATIONAL SEARCH REPORT

Inte ional Application No
PCT/US 99/23658

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L29/12 G05B19/418

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04L G05B

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 4 855 906 A (BURKE THOMAS J) 8 August 1989 (1989-08-08) column 3, line 42 - line 62; figure 1 column 4, line 26 - line 68 column 6, line 50 -column 8, line 68 column 10, line 18 - line 65 column 11, line 35 -column 12, line 50	1-7,9-12
Y		22-26, 28-30
A		8,13-16, 18,19,31

	-/--	

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
3 February 2000	10/02/2000

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Brichau, G

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Inte .onal Application No
PCT/US 99/23658

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 245 704 A (WEBER MARK S ET AL) 14 September 1993 (1993-09-14) column 1, line 28 - line 47 column 13, line 63 -column 17, line 44; claim 7	14-16, 18-20
Y		17,
A		22-26, 28-30 21,27,31
Y	US 5 771 174 A (MCCORMICK KEITH T ET AL) 23 June 1998 (1998-06-23) column 2, line 49 - line 56 column 4, line 13 - line 19 column 4, line 43 - line 67	17
A		3,4,9, 10,18, 27,28
A	US 4 888 726 A (STRUGER ODO J ET AL) 19 December 1989 (1989-12-19) column 1, line 7 - line 32 column 18, line 64 -column 19, line 43	8,13

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

Inte. onal Application No
PCT/US 99/23658

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 4855906 A	08-08-1989	NONE	
US 5245704 A	14-09-1993	NONE	
US 5771174 A	23-06-1998	CA 2239716 A	03-07-1997
		EP 0868703 A	07-10-1998
		WO 9723839 A	03-07-1997
US 4888726 A	19-12-1989	CA 1301940 A	26-05-1992